Deforming Autoencoders: Unsupervised Disentangling of Shape and Appearance

Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Güler, Dimitris Samaras, Nikos Paragios, Iasonas Kokkinos

Deforming Autoencoders: Unsupervised Disentangling of Shape and Appearance

Zhixin Shu[1] Mihir Sahasrabudhe[2] Rıza Alp Güler[2,3]
Dimitris Samaras[1] Nikos Paragios[4] Iasonas Kokkinos[5,6]

[1]Stony Brook University [2]CVN, CentraleSupélec [3]INRIA
[4]Therapanacea [5]University College London [6]Facebook AI Research

Abstract. In this work we introduce Deforming Autoencoders, a generative model for images that disentangles shape from appearance in an unsupervised manner. As in the deformable template paradigm, shape is represented as a deformation between a canonical coordinate system ('template') and an observed image, while appearance is modeled in 'canonical', template, coordinates, thus discarding variability due to deformations. We introduce novel techniques that allow this approach to be deployed in the setting of autoencoders and show that this method can be used for unsupervised group-wise image alignment. We show experiments with expression morphing in humans, hands, and digits, face manipulation, such as shape and appearance interpolation, as well as unsupervised landmark localization. A more powerful form of unsupervised disentangling becomes possible in template coordinates, allowing us to successfully decompose face images into shading and albedo, and further manipulate face images.

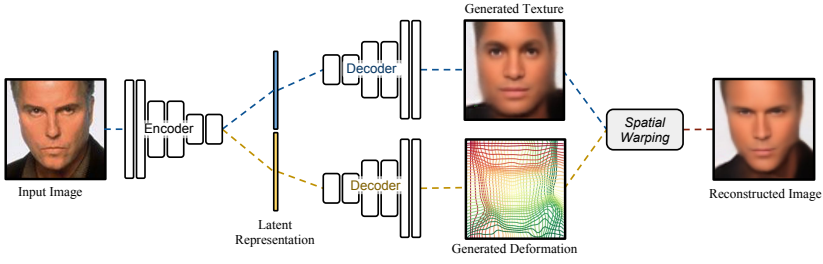

Fig. 1. Deforming Autoencoders follow the deformable template paradigm and model image generation through a cascade of appearance (or, 'texture') synthesis in a canonical coordinate system and a spatial deformation that warps the texture to the observed image coordinates. By keeping the latent vector for texture short the network is forced to model shape variability through the deformation branch, so as to minimize a reconstruction loss. This allows us to train a deep generative image model that disentangles shape and appearance in an entirely unsupervised manner.

1 Introduction

Disentangling factors of variation is important for the broader goal of controlling and understanding deep networks, but also for applications such as image manipulation through interpretable operations. Progress in the direction of disentangling the latent

space of deep generative models has facilitated the separation of latent image representations into dimensions that account for independent factors of variation, such as identity, illumination, normals, and spatial support [1–4], low-dimensional transformations, such as rotations, translation, or scaling, [5–7] or finer-levels of variation, including age, gender, wearing glasses, or other attributes e.g. [2, 8] for particular classes, such as faces.

Shape variation is more challenging as it amounts to a transformation of a function's domain, rather than its values. Even simple, supervised additive models of shape result in complex nonlinear optimization problems [9, 10]. Despite this challenge several works in the previous decade aimed at learning shape/appearance factorizations in an unsupervised manner, exploring groupwise image alignment, [11–14]. In the context of deep learning several works have aimed at incorporating deformations and alignment in a supervised setting, including Spatial Transformers [15], Deep Epitomic Networks [16], Deformable CNNs [17], Mass Displacement Networks [18], Mnemonic Descent [19], or Densereg [20]. These works have shown that one can improve the accuracy of both classification and localization tasks by injecting deformations and alignment within traditional CNN architectures.

Turning to unsupervised deep learning, even though most works focus on rigid, or low-dimensional parametric deformations, e.g. [5, 6], several works have attempted to incorporate richer non-rigid deformations within learning. A thread of works has been aimed at dynamically rerouting the processing of information within the network's graph based on the input, starting from neural computation arguments [21–23] and eventually translating into concrete algorithms, such as the 'capsule' works of [24, 25] that bind neurons on-the-fly. Still, these works lack a transparent, parametric handling of non-rigid deformations. Working on a more geometric direction, several works have recently aimed at recovering dense correspondences between pairs [26] or sets of RGB images, as e.g. in the recent works of [27, 28]. These works however do not have the notion of a reference coordinate system ('template') to which images can get mapped - this makes the image generation and manipulation harder. More recently, [29] use the equivariance principle in order to align sets of images to a common coordinate system, but do not develop this into a full-blown generative model of images.

Our work pushes the envelope of this line of research by following the deformable template paradigm [30, 31, 9, 32, 10]. In particular, we consider that object instances are obtained by deforming a prototypical object, or 'template', through dense, diffeomorphic deformation fields. This makes it possible to factor object variability within a category into variations that are associated to spatial transformations, generally linked to the object's 2D/3D shape, and variations that are associated to appearance (or, 'texture' in graphics), e.g. due to facial hair, skin color, or illumination. In particular we consider that both sources of variation can be modelled in terms of a low-dimensional latent code that is learnable in an unsupervised manner from images. We achieve disentangling by breaking this latent code into separate parts that are fed into separate decoder networks that deliver appearance and deformation estimates. Even though one could hope that a generic convolutional architecture will learn to represent such effects, we argue that explicitly injecting this inductive bias in a network can help with the training, while also yielding control over the generative process.

Our main contributions in this work can be summarized as follows:

First, we introduce the *Deforming Autoencoder* architecture, bringing together the deformable modeling paradigm with unsupervised deep learning. We treat the template-to-image correspondence task as that of predicting a smooth and invertible transformation. As shown in Fig. 1, our network predicts this transformation field alongside with the template-aligned appearance and subsequently deforms the synthesized appearance to generate an image similar to its input. This allows for a disentanglement of the shape and appearance parts of image generation by explicitly modelling the effects of image deformation during the decoding stage.

Second, we explore different ways in which deformations can be represented and predicted by the decoder. Instead of building a generic deformation model, we compose a global, affine deformation field, with a non-rigid field that is synthesized as a convolutional decoder network. We develop a method that allows us to constrain the synthesized field to be a diffeomorphism, namely an invertible and smooth transformation, and show that it simplifies training and improves accuracy. We also show that class-related information can be exploited, when available, to learn better deformation models: this yields sharper images and can be used to learn models that jointly account for multiple classes - e.g. all MNIST digits.

Third, we show that disentangling appearance from deformation comes with several advantages when it comes to modeling and manipulating images. By using disentangling we obtain clearly better synthesis results when manipulating images for tasks such as expression, pose or identity interpolation when compared to standard autoencoder architectures. Along the same lines, we show that accounting for deformations facilitates a further disentangling of the appearance components into an intrinsic, shading-albedo decomposition which completely fails when naively performed in the original image coordinates. This allows us to perform re-shading through simple operations on the latent shading coordinate space.

We complement these qualitative results with a quantitative analysis of the learned model in terms of landmark localization accuracy. We show that our method is not too far below supervised methods and outperforms with a margin the latest state-of-the-art works on self-supervised correspondence estimation [29], even though we never explicitly trained our network for correspondence estimation, but rather only aimed at reconstructing pixel intensities.

2 Deforming Autoencoders

Our architecture embodies the deformable template paradigm in an autoencoder architecture. The premise of our work is that image generation can be interpreted as the combination of two processes: a synthesis of appearance on a deformation-free coordinate system ('template'), followed by a subsequent deformation that introduces shape variability. Denoting by $T(\mathbf{p})$ the value of the synthesized appearance (or, texture) at coordinate $\mathbf{p} = (x, y)$ and by $W(\mathbf{p})$ the estimated deformation field, we consider that the observed image, $I(\mathbf{p})$ can be reconstructed as follows:

$$I(\mathbf{p}) \simeq T(W(\mathbf{p}), \tag{1}$$

namely the image appearance at position $\mathbf{p}$ is obtained by looking up the synthesized appearance at position $W(\mathbf{p})$. This is implemented in terms of a spatial transformer layer [15] that allows us to pass gradients through the warping process.

The appearance and deformation functions are synthesized by independent decoder networks. The inputs to the decoders are delivered by a joint encoder network that takes as input the observed image and delivers a low-dimensional latent representation, Z, of shape and appearance. This is split into two parts, $Z = [Z_T, Z_S]$ which feed into the appearance and shape networks respectively, providing us with a clear separation of shape and appearance.

2.1 Deformation field modeling

Rather than leave deformation modeling entirely to back-propagation, we use some domain knowledge to simplify and accelerate learning. The first observation is that global aspects can be expressed using low-dimensional linear models. We account for global deformations by an affine Spatial Transformer layer, that uses a six-dimensional input to synthesize a deformation field as an expansion on a fixed basis [15]. This means that the shape representation, Z_S described above is decomposed into two parts, Z_W, Z_A, where Z_A accounts for the affine, and Z_W for the non-rigid, learned part of the deformation field. These deformation fields are generated by separate decoders, and are *composed*, so that the affine transformation warps the detailed non-rigid warps to the image positions where they should apply. This is also a common decomposition in deformable models for faces [9, 10].

Turning to local deformation effects, we quickly realized that not every deformation field is plausible. Without appropriate regularization we would often obtain deformation fields that could expand small areas to occupy whole regions, and/or would be non-diffeomorphic, meaning that the deformation could spread a connected texture pattern to a disconnected image area (Figure 2-(f)).

To prevent this problem, instead of making the shape decoder CNN directly predict the local warping field $W(\mathbf{p}) = (W_x(x, y), W_y(x, y))$, we consider a 'differential decoder' that generates the spatial gradient of the warping field: $\nabla_x W_x$ and $\nabla_y W_y$, where ∇_c denotes the $c - th$ component of the spatial gradient vector. These two quantities measure the displacement of consecutive pixels - for instance $\nabla_x W_x = 1$ amounts to translation in the horizontal axis, $\nabla_x W_x = 2$ amounts to horizontal shifting by a size of 2, while $\nabla_x W_x = -1$ amounts to left-right flipping; a similar behavior is associated with $\nabla_y W_y$ in the vertical axis. We note that global rotations are handled by the affine warping field, and the $\nabla_x W_y, \nabla_y W_x$ are associated with small local rotations of minor importance - we therefore focus on $\nabla_x W_x, \nabla_y W_y$.

Having access to these two values gives us a handle on the deformation field, since we can prevent folding/excessive stretching by controlling $\nabla_x W_x, \nabla_y W_y$.

In particular, we pass the outputs of our differential decoder through a Rectified Linear Unit (ReLU) module, which enforces positive horizontal offsets on horizontally adjacent pixels, and positive vertical offsets on vertically adjacent pixels. We subsequently apply a spatial integration layer, implemented in terms of a fixed network layer, on top of the output of the ReLU layer to reconstruct the warping field

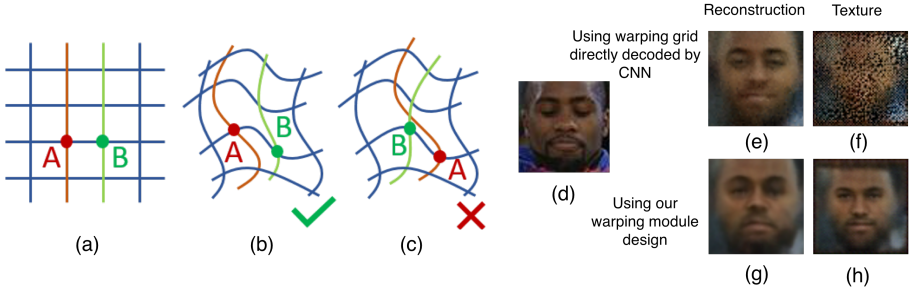

Fig. 2. Our warping module design only permits locally consistent warping, as shown in (b), while the flipping of relative pixel positions, as shown in (c), is not allowed by design. To achieve this, we let the deformation decoder predict the horizontal and vertical increments of the deformation ($\nabla_x W$ and $\nabla_y W$, respectively) and use a ReLU transfer function to remove local flips, caused by going back in the vertical or horizontal direction. A spatial integral module is subsequently applied to generate the grid. This simple mechanism serves as an effective constraint for the deformation generation process, while allowing us to model free-form/non-rigid local deformation.

from its spatial gradient. By doing so, the new deformation module enforces the generation of smooth and regular warping fields that avoid self-crossings. In practice we found that also clipping the decoded offsets by a maximal value significantly eases the training, which amounts to replacing the ReLU layer, $\mathrm{ReLU}(x) = \max(x, 0)$ with a $\mathrm{HardTanh}_{0,\delta}(x) = \min(\max(x, 0, \delta)$ layer. In our experiments, we set $\delta = 5/W$ where W denotes the number of pixels along one dimension of the image.

2.2 Class-aware Deforming Autoencoder

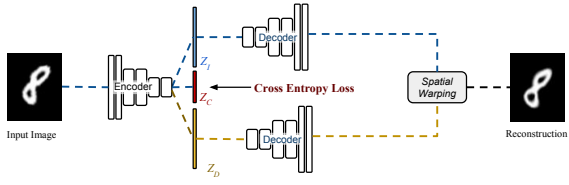

Fig. 3. A *class-aware* model can account for multi-modal deformation distributions by utilizing class information. Introducing a classification loss into latent space helps the model learn a better representation of the input as demonstrated on MNIST.

We can require our network's latent representation to be predictive of not only shape and appearance, but also of instance class, if that is available during training. We note that this information, being discrete may be easier to acquire than the actual deformation field, which would require manual landmark annotation. For instance, for faces such discrete information could represent the expression or a person's identity.

In particular we consider that the latent representation can be decomposed as follows: $Z = [Z_T, Z_C, Z_S]$, where Z_T, Z_S are as previously the appearance- and shape-related parts of the representation, respectively, while Z_C is fed as input to a sub-network trained to predict the class associated with the input image. Apart from assisting the classification task, the latent vector Z_C is fed into both the appearance and shape decoders. Intuitively this allows our decoder network to learn a mixture model that is conditioned on class information, rather than treating the joint, multi-modal distribution through a monolithic model. Even though the class label is only used during training, and not for reconstruction, our experimental results show that a network trained with class supervision can deliver more accurate synthesis results.

2.3 Intrinsic Deforming Autoencoder: Deformation, Albedo and Shading Decomposition

Having outlined Deforming Autoencoders, we now use a Deforming Autoencoder to model complex physical image signals, such as illumination effects, without a supervision signal. For this we design the Intrinsic Deforming-Autoencoder, named Intrinsic-DAE to model shading and albedo for in-the-wild face images. As shown in Fig. 4-(a), we introduce two separate decoders for shading S and albedo A, each of with has the same structure as the original texture decoder. The texture is computed by $T = S \circ A$ where $\circ$ denotes the Hadamard product.

In order to model the physical properties of shading and albedo, we follow the intrinsic decomposition regularization loss used in [2]: we apply the L2 smoothness loss on ∇S, meaning that shading is expected to be smooth, while leaving albedo unconstrained. As shown in Fig. 4 and more extensively in the experimental results section, when used in tandem with an Deforming Autoencoder this allows us to successfully decompose of face image into shape, albedo, and shading components, while a standard Autoencoder completely fails at decomposing unaligned images into shading and albedo.

2.4 Training

Our objective function is formed as the sum of three losses, combining the reconstruction error with the regularization terms required for the modules described above. Concretely, the loss of the deforming autoencoder can be written as

$$E_{\text{DAE}} = E_{\text{Reconstruction}} + E_{\text{Warp}}, \tag{2}$$

where the reconstruction loss is defined as the standard ℓ_2 loss

$$E_{\text{Reconstruction}} = \|I_{\text{Output}} - I_{\text{Input}}\|^2, \tag{3}$$

and the warping loss is decomposed as follows:

$$E_{\text{Warp}} = E_{\text{Smooth}} + E_{\text{BiasReduce}}. \tag{4}$$

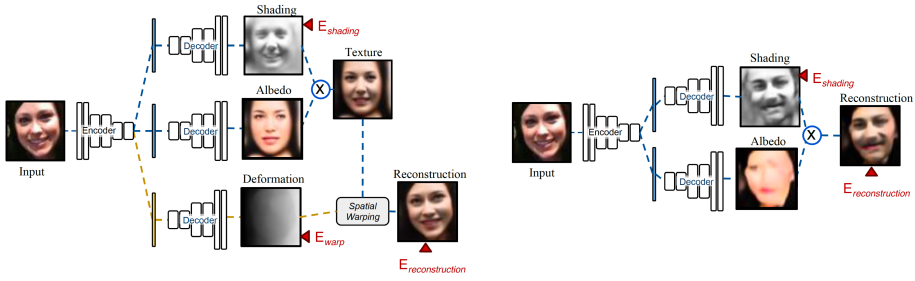

Fig. 4. Autoencoders with intrinsic decomposition. (a) Deforming Autoencoder with intrinsic decomposition (Intrinsic-DAE): we model the texture by the Hadamard product of shading and albedo components, each of which is decoded by an individual decoder. The texture is subsequently warped by the predicted deformation field. (b) A plain autoencoder with intrinsic decomposition. Both networks are trained with reconstruction loss ($E_{\text{Reconstruction}}$) on the final output and regularization losses on shading (E_{Shade}) and deformation (E_{Warp}), if it exists.

In particular the smoothness cost, E_{smooth}, penalizes quickly-changing deformations encoded by the local warping field. It is measured in terms of the total variation norm of the horizontal and vertical differential warping fields, and is given by

$$E_{\text{Smooth}} = \lambda_1 \left(\|\nabla W_x(x,y)\|_1 + \|\nabla W_y(x,y)\|_1 \right), \tag{5}$$

where $\lambda_1 = 1e-6$. Finally, $E_{\text{BiasReduce}}$ is a regularization on (1) the affine parameters defined as the L2-distance between S_A and S_0, S_0 being the identity affine transform and (2) the average of the deformation grid for a random batch of training data being close to identity mapping grid:

$$E_{\text{BiasReduce}} = \lambda_2 \|S_A - S_0\|^2 + \lambda_2' \|\bar{W} - W_0\|^2, \tag{6}$$

where $\lambda_2 = \lambda_2' = 0.01$. $\bar{W}$ denotes the average deformation grid of a mini-batch of training data and W_0 denotes an identity mapping grid. In the class-aware variant described in Sec. 2.2 we augment the loss above with the cross-entropy loss evaluated on the classification network's outputs.

For Intrinsic-DAE, we add the following objective function in training: $E_{\text{Shade}} = \lambda_3 \|\nabla S\|^2$ where λ_3 =1e-6.

We experiment with two types of architectures; the majority of our results are obtained with a standard auto-encoder architecture, where both encoder and decoders are CNNs with standard convolution-BatchNorm-ReLU blocks. The number of filters and the texture bottleneck capacity can vary per experiment, image resolution, and dataset, as detailed in the Appendix A.

Follow the recent work on densely connected convolutional networks [33], we have also experimented with incorporating dense connections into our encoder and decoders architectures respectively (no skip connections over the bottleneck layer for latent representations). In particular, we follow the architecture of DenseNet-121, but without the 1×1 convolutional layers inside each dense block. These have been shown to better

exploit larger datasets, as indicated in the quantitative analysis of unsupervised face alignment. We call this version of the deforming autoencoder Dense-DAE.

3 Experiments

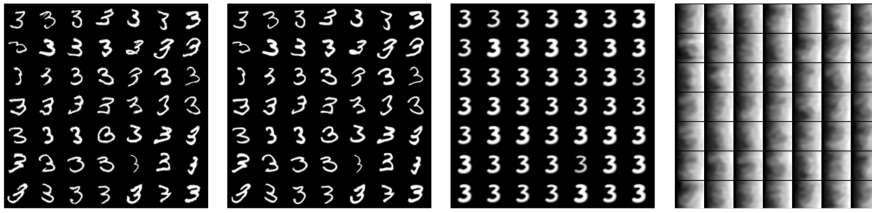

Fig. 5. Unsupervised deformation-appearance disentangling on a single MNIST digit. Our network learns to reconstruct the input image while automatically deriving a canonical appearance for the input image class. In this experiment, the dimension of the latent representation for appearance Z_T is 1.

To demonstrate the properties of our deformation disentangling network, we conduct experiments on the following three datasets:

- Deformed MNIST. A synthetic dataset designed specifically to explore the deformation modelling power of our network. Deformed MNIST consists of handwritten MNIST images randomly distorted using a mixture of sinusoidal waveforms.
- MUG facial expression dataset [34]. This dataset consists of videos of individuals performing facial expressions, with simple blue background and minor translation. The dataset also offers frames from the videos, classified according to the facial expression, as well as the subject.
- Faces-in-the-wild dataset: MAFL [35] and CelebA [36]. These datasets consist of uncontrolled "in-the-wild" faces with variability in pose, illumination, expression, age, etc.

Using these datasets we experimentally explored the ability of the unsupervised appearance-shape (or texture-deformation) disentangling network on 1) unsupervised image alignment/appearance inference; 2) learning semantically meaningful manifolds for shape and appearance; 3) decomposition into illumination intrinsics (shading, albedo); 4) unsupervised landmark detection, as detailed below. We intend to make all of the code of our system publicly available in order to facilitate the reproduction of our results.

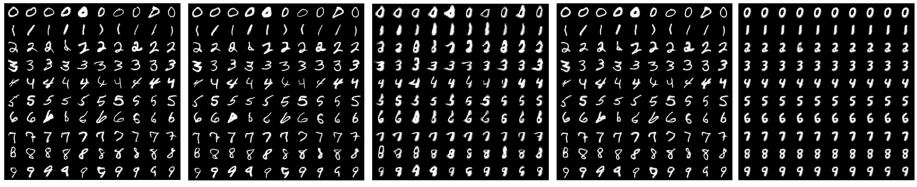

Fig. 6. Class-aware Deforming Autoencoders effectively model the appearance and deformation for multi-class data.

3.1 Unsupervised Appearance Inference

We first use our network to model canonical appearance and deformation for single category objects. For this purpose, we demonstrate the results in the MNIST and MUG facial expression datasets (Fig. 5, 6, 7).

We observe that by heavily limiting the size of Z_T (1 in Fig. 5 and 0 in Fig. 7), we can successfully infer a canonical appearance for such a class. In Fig. 5, all different types of handwritten digits '3' are aligned to a simple canonical shape. In Fig. 7, by limiting the dimension of Z_T to 0, the network learns to encode a single texture image for all expressions, and successfully distills expression-related information exclusively in the shape space. In Fig. 7-(b) we show that by interpolating the learned latent representations, we can generate meaningful shape interpolations that mimic facial expressions.

In cases where data has a multi-modal distribution exhibiting multiple different canonical appearances, e.g., multi-class MNIST digit images, learning a single appearance is less meaningful and often challenging (Fig. 6-(b)). In such cases, utilizing class information (Sec. 2.2) significantly improves the quality of multi-modal appearance learning (Fig. 6-(d)). As the network learns to classify the images implicitly in its latent space, it learns to generate a single canonical appearance for each class. Misclassified data will be decoded into an incorrect class: the image at position (2,4) in Fig. 6-(c,d) is interpreted as a 6.

We now demonstrate the effectiveness of texture inference using our network on in-the-wild human faces. Using the MAFL face dataset, we show that our network is able to align the faces to a common texture space under various poses, illumination conditions, or facial expressions (Fig. 10)-(d). The aligned textures retain the information of the input image such as lighting, gender, and facial hair, without a relevant supervision training signal. We further demonstrate the alignment on the 11k Hands dataset [37], where we align palmar images of the left hand of several subjects 8. This property of our network is especially useful for applications such as computer graphics, where establishing correspondences (UV map) between a class of objects is important but usually difficult.

3.2 Autoencoders vs. Deforming Autoencoders

We show the ability of our network to learn meaningful deformation representations without supervision. We compare our disentangling network with a plain auto-encoder

(a) Learning single template from image sequence

(b) Expression interpolation

Fig. 7. Experiment on MUG dataset of face expressions: (a) With 0-length Z_T, Deforming Autoencoders learn a single texture (row 3) from a subject in the MUG facial expression dataset. By doing so, the subject's facial expression is encoded only in the deformation domain. (b): Our network is able to disentangle the facial expression deformation and encode this information in a meaningful latent representation. By interpolating the latent deformation representation from the source (in orange) to the target (in blue), it generates sharp images and a smooth deformation interpolation between expressions as shown in each row.

(a) (b) (c) (d) (e)

Fig. 8. Unsupervised alignment on images of palms of left hands. (a) The input images; (b) reconstructed images; (c) texture images warped with the average of the decoded deformation; (d) the average input image; and (e) the average texture.

(Fig. 9). Contrary to our network which disentangles an image into a template texture and a deformation field, the auto-encoder is trained to encode all of the image in a single latent representation, i.e., the bottleneck.

We train both networks in the MAFL faces-in-the-wild dataset. To evaluate the learned representation, we conduct manifold traversal (i.e., latent representation interpolation) between two randomly sampled face images: given a source face image I^s and a target image I^t, we first compute their latent representations Zs. We use $Z_T(I^s)$ and $Z_S(I^s)$ to denote the latent representations in our network for I^s, and $Z_{ae}(I^s)$ for the latent representation learned by a plain autoencoder. We then conduct linear interpolation on Z, between Z^s and Z^t: $Z^\lambda = \lambda Z^s + (1-\lambda)Z^t$. We subsequently reconstruct the image I^λ from Z^λ using the corresponding decoder(s), as shown in Fig. 9.

By traversing the learned deformation representation only, we can change the shape and pose of a face while maintaining its texture (Fig. 9-(1)); interpolating the texture representation results in pose-aligned texture transfer (Fig. 9-(2)); traversing on both representations will generate a smooth deformation from one image to another (Fig. 9-(3,5,7)). Compared to the interpolation using the autoencoder (Fig. 9-(4,6,8)), which

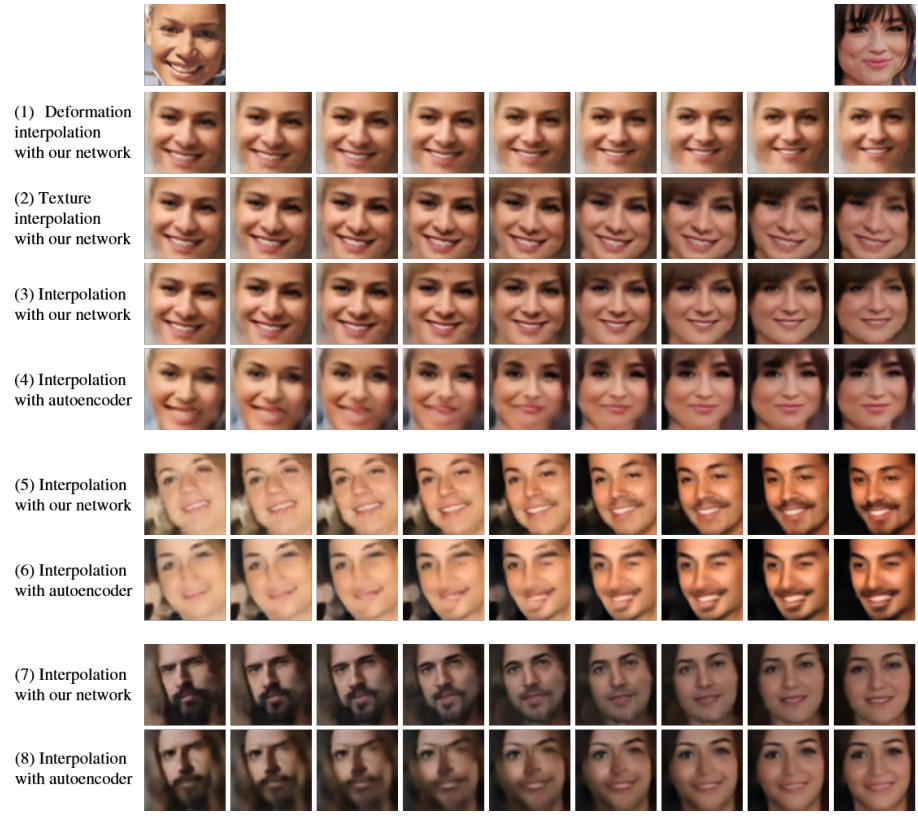

Fig. 9. Latent representation interpolation: we embed a face image in the latent space provided by an encoder network trained on the MAFL dataset. Our network disentangles the texture and deformation in the respective parts of the latent representation vector, allowing a meaningful interpolation between images. Interpolating the deformation-specific part of the latent representation changes the face shape and pose (1); interpolating the latent representation for texture will generate a pose-aligned texture transfer between the images (2); traversing both latent representations will generate smooth and sharp image deformations (3,5,7). In contrast, when using a standard auto-encoder (4,6,8) such an interpolation often yields artifacts. For more results, please see Figure 18,19 in Appendix.

often exhibits artifacts, our traversal stays on the semantic manifold of faces and generates sharp facial features.

3.3 Intrinsic Deforming Autoencoders

Having demonstrated the disentanglement abilities of Deforming Autoencoders, we now explore the disentanglement capabilities of Intrinsic-DAE described in Sec. 2.3. Using only the E_{DA} and regularization losses, the Intrinsic-DAE is able to generate convincing shading and albedo estimates without direct supervision (Fig. 10-(b) to (g)).

Without the "learning-to-align" property, a baseline autoencoder structure with an intrinsic decomposition design (Fig. 4-(b)) cannot decompose the image into plausible shading and albedo components (Fig. 10-(h),(i),(j)).

In addition, we show that by manipulating the learned latent representation of S, Intrinsic-DAE allows us to simulate illumination effects for face images, such as interpolating lighting directions (Fig. 11).

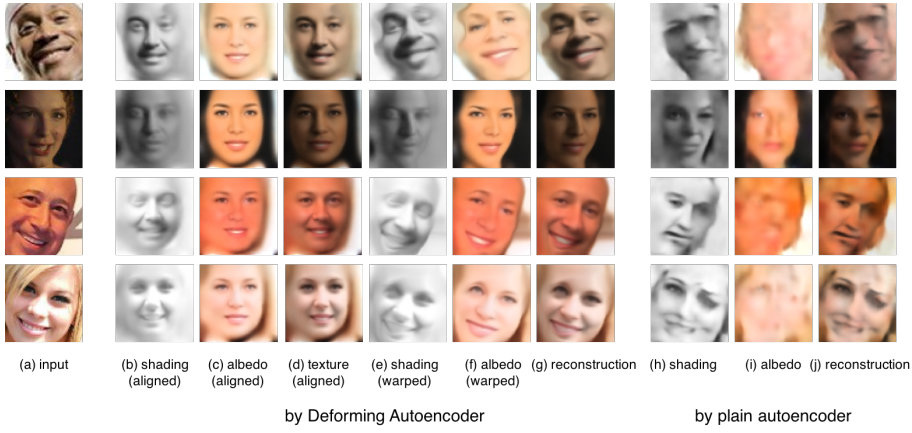

Fig. 10. Unsupervised intrinsic decompostion with Deforming Autoencoders (Intrinsic-DAE). Thanks to the "automatic dense aligment" property of DAE, shading and albedo are faithfully separated (e,f) by the intrinsic decomposition loss. Shading (b) and albedo (c) are learned in an unsupervised manner in the densely aligned canonical space. With the deformation field also learned without supervision, we can recover the intrinsic image components for the original shape and viewpoint (e,f). Without dense alignment, the intrinsic decomposition loss fails to decompose shading and albedo (h,i,j).

Training with $L2$ reconstruction losses, autoencoder-like architectures are prone to generating smooth images which lack visual realism (Fig. 10). Inspired by the success of generative adversarial networks (GANs) [38], we follow previous work [2] where an adversarial loss is adopted to generate visually realistic images: we train the Intrinsic-DAE with an extra adversarial loss term $E_{\text{Adversarial}}$ applied on the final output. The loss function becomes:

$$E_{\text{Intrinsic-DAE}} = E_{\text{Reconstruction}} + E_{\text{Warp}} + \lambda_4 E_{\text{Adversarial}}. \tag{7}$$

In practice, we apply a PatchGAN [39, 40] as the discriminator and set $\lambda_4 = 0.1$. We found that the adversarial loss improves the visual sharpness of the reconstruction while the deformation, shading are still successfully disentangled (Fig. 12).

3.4 Unsupervised alignment evaluation

Having qualitatively analyzed the disentanglement capabilities of our networks, we now turn to quantifying their performance on the task of unsupervised image alignment. We

Fig. 11. Lighting interpolation with Intrinsic-DAE. With latent representations learned in an unsupervised manner for shading, albedo, and defomation, the DAE allows us to simulate smooth transitions of the lighting direction. In this example, we interpolate the latent representation of the shading from source (lit from the left) to target (mirrored source, hence lit from the right). The network generates smooth lighting transitions, without explicitly learning geometry, as shown in shading (1) and texture (2). Together with the learned deformation of the source image, DAE enables the relighting of the face in its original pose (3).

report the performance of our face DAE's alignment on landmark detection on face images, specifically, the eyes, the nose, and corners of the mouth. We report performance on the MAFL dataset, which contains manually annotated landmark locations for 19,000 training and 1,000 test images. In our experiments, we use a model trained on the CelebA dataset without any form of supervision to estimate deformation fields on the MAFL training set. Following the evaluation protocol of the work that we directly compare to [29], we train a landmark regressor post-hoc on these deformation fields using the provided annotations. We use landmark locations from the MAFL training set as training data for this regressor, but do not pass gradients to the Deforming Autoencoder, which thereby remains fixed to the model learned without supervision. The regressor is a 2-layer fully-connected neural network. Its inputs are flattened deformation fields (vectors of size $64 \times 64 \times 2$), which are provided as input to a 100-dimensional hidden layer, followed by a ReLU and a 10-D output layer to predict the spatial coordinates $((x, y))$ for five landmarks corresponding to the eyes, nose, and mouth corner landmarks. We use L1 loss as the objective function for this regression task.

In testing, we predict landmark locations using the trained regressor and the deformation fields on the MAFL test set. In Table 1 we report the mean error in landmark localization as a percentage of the inter-ocular distance. As the deformation field determines the alignment in the texture space, it serves as an effective mapping between landmark locations on the aligned texture and those on the original, unaligned faces. Hence, the mean error we report directly quantifies the quality of the (unsupervised) face alignment.

In Table 2 we compare with the results of the best current method for semi-supervised image registration [29]. We observe that by better modeling of the deformation space we quickly bridge the gap in performance, even though we never explicitly trained to learn correspondences.

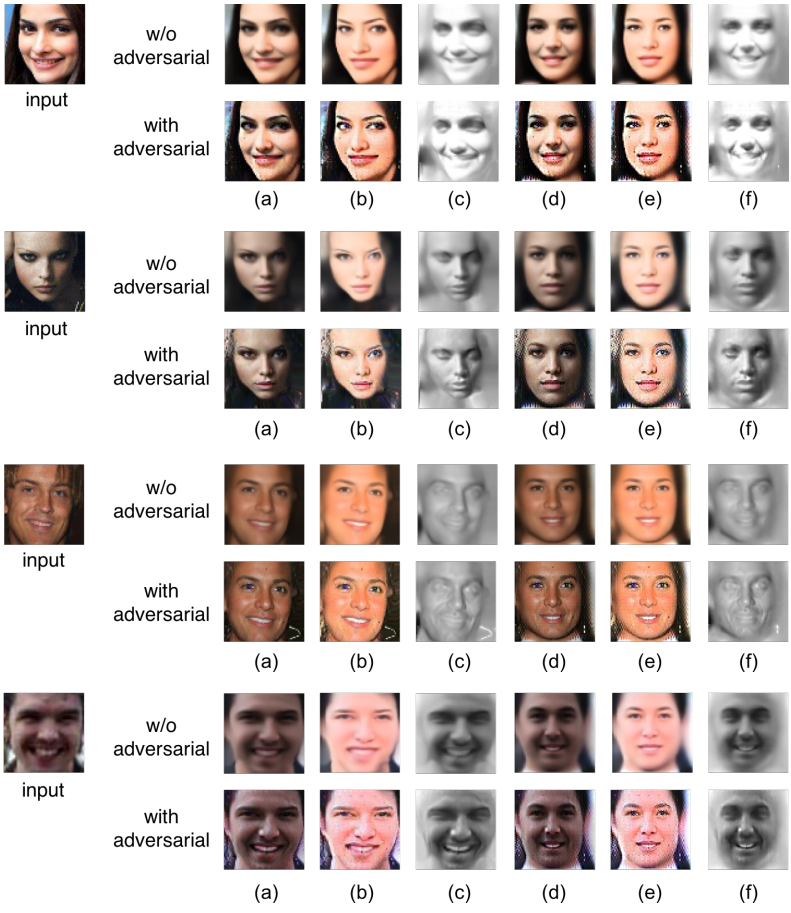

Fig. 12. Intrinsic-DAE with an adversarial loss: (a/d) reconstruction (b/e) albedo, (c/f) shading, in image and template coordinates, respectively. Applying an adversarial loss to the final output results improves the visual quality of the image reconstruction (a) of Intrinsic-DAE, while the deformation, albedo, and shading can still be successfully disentangled.

A, MAFL	I, MAFL	$A + I$, MAFL	$A + I$, CelebA	$A + I$, CelebA, with Regressor
14.13	9.89	8.50	7.54	5.96

Table 1. Improvement in landmark localization errors on the MAFL test set as we add new types of deformation and new data. In the table, A indicates a model which uses the affine transformation, I indicates one with the integral transformation, whereas MAFL and CelebA denote which dataset the deforming autoencoder was trained on. For columns 1 to 4, we manually annotate landmarks on the average texture image, while for column 5, we train a regressor on the deformation fields to predict them. In all experiments, each latent vector in the DAE is of size 32.

4 Conclusion and Future Work

In this paper we have developed deep autoencoders that can disentangle shape and appearance in latent representation space. We have shown that this method can be used

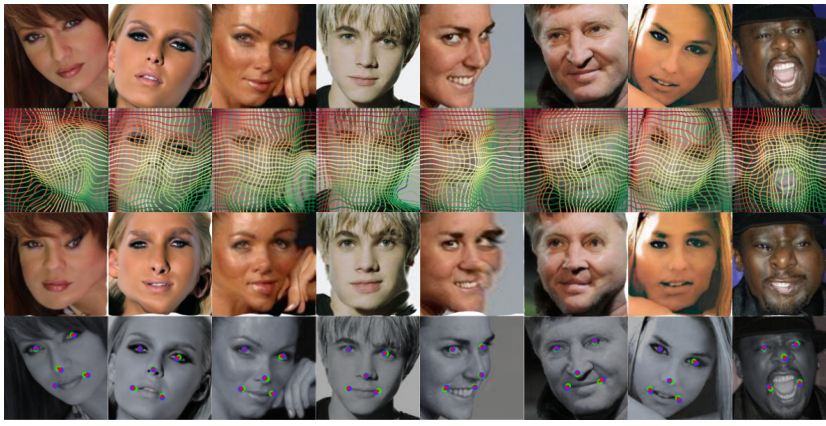

Fig. 13. *1st row*: Sample images from the MAFL test set; *2nd row*: Estimated deformation grid; and *3rd row*: Image reverse-transformed to texture space *4th row*: semantic landmark locations (green: ground truth landmark locations, blue: estimated landmark locations, red: error lines).

DAE						Dense-DAE			TCDCN[41]	Thewlis et al.[29]
32-NR	32-Res	16	32	64	96	16	64	96		
10.24	9.93	5.71	5.96	5.70	6.46	6.85	5.50	**5.45**	7.95	5.83

Table 2. Mean error on unsupervised landmark detection on the MAFL test set, expressed as a percentage of the inter-ocular distance: modeling non-rigid deformations clearly reduces error more than just modeling affine ones. DAE and Dense-DAE denote two flavours of the deforming autoencoder - with and without dense convolutional connections, respectively. Under DAE and Dense-DAE we specify the size of each latent vector in the deforming autoencoder. *NR* signifies training without regularization on the estimated deformations, while *Res* signifies training by estimating the residual deformation grid instead of the integral. Our results clearly outperform the self-supervised method of [29] trained specifically for establishing correspondences.

for unsupervised groupwise image alignment. Our experiments with expression morphing in humans, image manipulation, such as shape and appearance interpolation, as well as unsupervised landmark localization, show the generality of our approach. We have shown that bringing images in a canonical coordinate system allows for a more extensive form of image disentangling, facilitating the estimation of decompositions into shape, albedo and shading without any form of supervision. We expect that this will lead in the future to a full-fledged disentanglement into normals, illumination, and 3D geometry.

5 Acknowledgment

This work was supported by a gift from Adobe, NSF grants CNS-1718014 and DMS 1737876, the Partner University Fund, and the SUNY2020 Infrastructure Transportation Security Center.

References

1. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS. (2016)
2. Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., Samaras, D.: Neural face editing with intrinsic image disentangling. In: CVPR. (2017)
3. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Interpretable transformations with encoder-decoder networks. In: CVPR. (2017)
4. Sengupta, S., Kanazawa, A., Castillo, C.D., Jacobs, D.: Sfsnet: Learning shape, reflectance and illuminance of faces in the wild. arXiv preprint arXiv:1712.01261 (2017)
5. Memisevic, R., Hinton, G.E.: Learning to represent spatial transformations with factored higher-order boltzmann machines. Neural Computation (2010)
6. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic networks: Deep translation and rotation equivariance. (2016)
7. Park, E., Yang, J., Yumer, E., Ceylan, D., Berg, A.C.: Transformation-grounded image generation network for novel 3d view synthesis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2017) 702–711
8. Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., Ranzato, M.: Fader networks: Manipulating images by sliding attributes. CoRR **abs/1706.00409** (2017)
9. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: European conference on computer vision, Springer (1998)
10. Matthews, I., Baker, S.: Active appearance models revisited. IJCV (2004)
11. Learned-Miller, E.G.: Data driven image models through continuous joint alignment. PAMI (2006)
12. Kokkinos, I., Yuille, A.L.: Unsupervised learning of object deformation models. In: ICCV. (2007)
13. Frey, B.J., Jojic, N.: Transformation-invariant clustering using the EM algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **25**(1) (2003) 1–17
14. Jojic, N., Frey, B.J., Kannan, A.: Epitomic analysis of appearance and shape. In: 9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France. (2003) 34–43
15. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. CoRR **abs/1506.02025** (2015)
16. Papandreou, G., Kokkinos, I., Savalle, P.: Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In: CVPR. (2015)
17. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. (2017)
18. Neverova, N., Kokkinos, I.: Mass displacement networks. Arxiv (2017)
19. Trigeorgis, G., Snape, P., Nicolaou, M.A., Antonakos, E., Zafeiriou, S.: Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In: Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition. (2016)
20. Güler, R.A., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., Kokkinos, I.: Densereg: Fully convolutional dense shape regression in-the-wild. CVPR (2017)
21. Hinton, G.E.: A parallel computation that assigns canonical object-based frames of reference. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981. (1981) 683–685
22. Olshausen, B.A., Anderson, C.H., Essen, D.C.V.: A multiscale dynamic routing circuit for forming size- and position-invariant object representations. Journal of Computational Neuroscience **2**(1) (1995) 45–62

23. Malsburg, C.: The correlation theory of brain function. In: Internal Report 81-2. Gottingen Max-Planck-Institute for Biophysical Chemistry. (1981)
24. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I. (2011) 44–51
25. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. CoRR **abs/1710.09829** (2017)
26. Bristow, H., Valmadre, J., Lucey, S.: Dense semantic correspondence where every pixel is a classifier. In: ICCV. (2015)
27. Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: CVPR. (2016)
28. Gaur, U., Manjunath, B.S.: Weakly supervised manifold learning for dense semantic object correspondence. In: ICCV. (2017)
29. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised object learning from dense equivariant image labelling. (2017)
30. Amit, Y., Grenander, U., Piccioni, M.: Structural image restoration through deformable templates. Journal of the American Statistical Association **86**(414) (1991)
31. Yuille, A.L.: Deformable templates for face recognition. Journal of Cognitive Neuroscience **3**(1) (1991)
32. Blanz, V.T., Vetter, T.: Face recognition based on fitting a 3D morphable model. **25**(9) (2003) 1063–1074
33. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017)
34. 11th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2010, Desenzano del Garda, Italy, April 12-14, 2010, IEEE (2010)
35. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI. (2014) 94–108
36. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV). (2015)
37. Afifi, M.: Gender recognition and biometric identification using a large dataset of hand images. CoRR **abs/1711.04322** (2017)
38. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
39. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: European Conference on Computer Vision, Springer (2016) 702–716
40. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arxiv (2016)
41. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Learning deep representation for face alignment with auxiliary attributes. IEEE transactions on pattern analysis and machine intelligence **38**(5) (2016) 918–930

A Architectural Details

A.1 Convolutional Encoders and Decoders

In our experiments, where input images are of size $64 \times 64 \times$ Nc (Nc is 1 for MNIST and 3 for faces), we use identical architectures for convolutional encoders and decoders.

The encoder architecture is

```
Conv(32)-LeakyReLU-Conv(64)-BN-LeakyReLU-Conv(128)->
 ->BN-LeakyReLU-Conv(256)-BN-LeakyReLU-Conv(Nz)->
 ->Sigmoid;
```

while the decoder architecture is

```
ConvT(256)-BN-ReLU-ConvT(128)-BN-ReLU-ConvT(64)->
 ->BN-ReLU-ConvT(32)-BN-ReLU-ConvT(32)-BN-ReLU-ConvT(Nc)->
 ->Threshold(0,1),
```

where

- `Conv(n)`: convolution layer with n output feature map;
- `ConvT(n)`: transposed convolution (deconvolution) layer with n output feature map;
- `BN`: batch normalization layer
- `Nz`: latent representation dimension
- `Nc`: number of output image channel

A.2 DenseNet-style Encoders and Decoders

For DenseNet-style architectures, we employ dense convolutional connections. The architecture for the encoder is

```
BN-ReLU-Conv(32)-DBE(32,6)-TBE(32,64,2)->
 ->DBE(64,12)-TBE(64,128,2)-DBE(128,24)-TBE(128,256,2)->
 ->DBE(256,16)-TBE(256,Nz,4)-Sigmoid;
```

whereas the architecture for the decoder is

```
BN-Tanh-ConvT(256)-DBD(256,16)-TBD(256,128)->
 ->DBD(128,24)-TBD(128,64)-DBD(64,12)-TBD(64,32)->
 ->DBD(32,6)-TBD(32,32)-BN-Tanh-ConvT(Nc)-Threshold(0,1),
```

where

- `DBE(n,k)`: A dense encoder block with k 3×3 convolutions with n channels.
- `TBE(m,n,p)`: An encoder transition block of 1×1 convolutions with m input channels and n output channels. Also includes a max-pooling operation of size p.
- `DBD(n,k)`: A dense decoder block with k 3×3 transposed convolution operations with n channels.
- `TBD(m,n)`: A decoder transition block of 4×4 convolutions, stride of 2 and padding of 1. It has m input channels, and n output channels.

We describe the tensor sizes for intermediate convolution operations in Tables 3 and 4.

Conv Encoder		Conv Decoder	
Output Size	Operation	Output size	Operation
$32 \times 32 \times 32$	4×4 Conv(32)	$4 \times 4 \times 256$	4×4 ConvT(256)
$16 \times 16 \times 64$	4×4 Conv(64)	$4 \times 4 \times 128$	4×4 ConvT(128)
$8 \times 8 \times 128$	4×4 Conv(128)	$4 \times 4 \times 64$	4×4 ConvT(64)
$4 \times 4 \times 256$	4×4 Conv(256)	$4 \times 4 \times 32$	4×4 ConvT(32)
Nz	4×4 Conv(Nz)	$4 \times 4 \times 32$	4×4 ConvT(32)
		$4 \times 4 \times$Nc	4×4 ConvT(Nc)

Table 3. Tensor sizes for intermediate convolutional operations in the convolutional encoder and decoder architectures. The output shape denoted $h \times w \times C$, where h and w are height and width of the feature maps, respectively, and C is the number of channels.

Dense Conv Encoder		Dense Conv Decoder	
Output Size	Operation	Output size	Operation
$32 \times 32 \times 32$	4×4 Conv(32)	$4 \times 4 \times 256$	4×4 ConvT(256)
$32 \times 32 \times 32$	DBE(32,6)	$4 \times 4 \times 256$	DBD(256,16)
$16 \times 16 \times 64$	TBE(32,64,2)	$8 \times 8 \times 128$	TBD(256,128)
$16 \times 16 \times 64$	DBE(64,12)	$8 \times 8 \times 128$	DBD(128,24)
$8 \times 8 \times 128$	TBE(64,128,2)	$16 \times 16 \times 64$	TBD(128,64)
$8 \times 8 \times 128$	DBE(128,24)	$16 \times 16 \times 64$	DBD(64,12)
$4 \times 4 \times 256$	TBE(128,256,2)	$32 \times 32 \times 32$	TBD(64,32)
$4 \times 4 \times 256$	DBE(256,16)	$32 \times 32 \times 32$	DBD(32,6)
Nz	TBE(256,Nz,4)	$64 \times 64 \times 32$	TBD(32,32)
		$64 \times 64 \times$Nc	3×3 ConvT(Nc)

Table 4. Tensor sizes for intermediate convolutional operations in the dense encoder and decoder architectures. The output shape denoted $h \times w \times C$, where h and w are height and width of the feature maps, respectively, and C is the number of channels.

B Ablation Study

B.1 Dimension of Z_T

In this section, we show experimental results on single deformed MNIST images of the digit 3 (Figure 14) as well as in-the-wild faces (without masking) from the MAFL dataset (Figure 15) to demonstrate the effect of varying the dimension of Z_T.

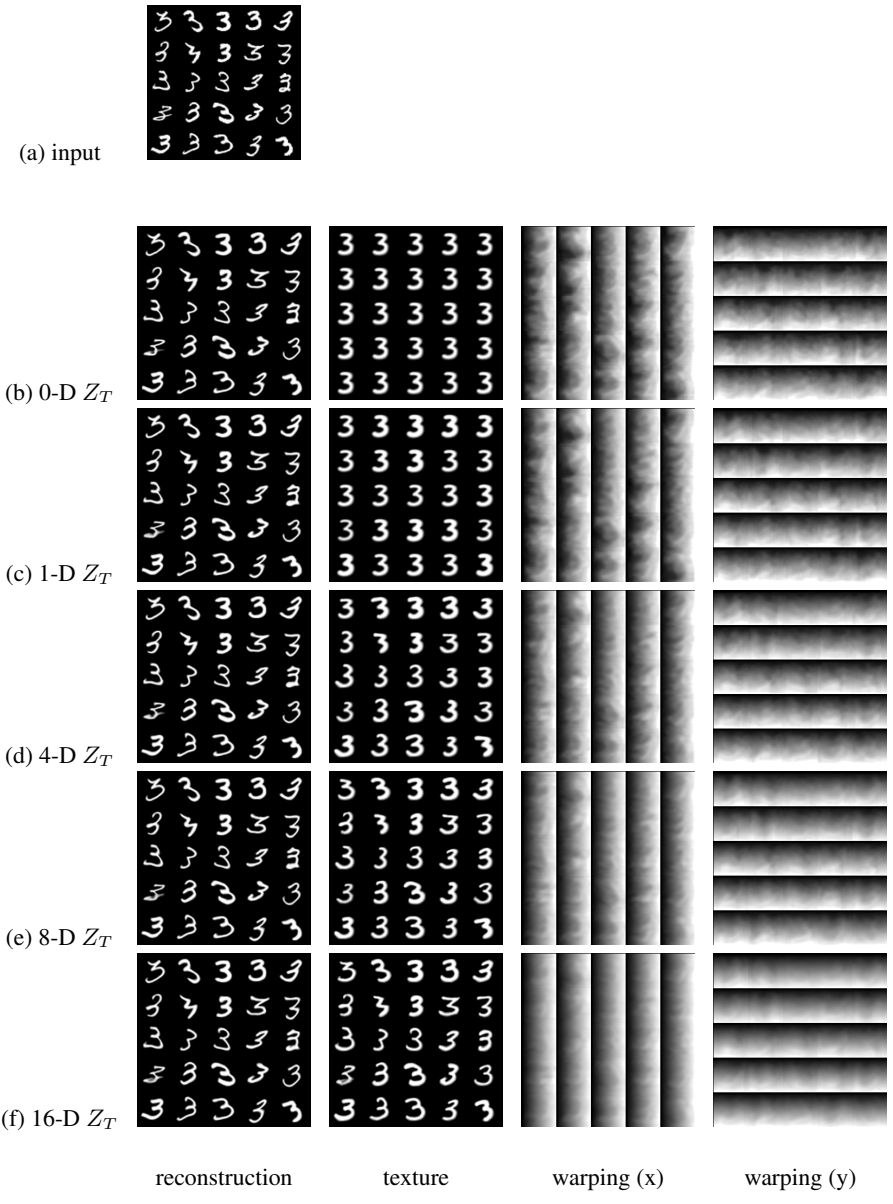

Fig. 14. Effect of varying the dimensionality of the latent vector for the texture encoding, Z_T: The dimension of Z_T is 0 for (b), 1 for (c), 4 for (d), 8 for (e), 16 for (f). Z_W is fixed to 128. When Z_T is 0-Dimensional, the texture decoder is forced to generate an identical texture for every image (b). When we increase the dimension of Z_T to 1, the texture decoder learns to align the pose (c) with varying stroke width. When further increasing the dimension of Z_T, the network learns a more diverse texture map for each image (d, e, f).

Fig. 15. Effect of varying the dimensionality of the latent vector for the texture envoding, Z_T, on the MAFL face dataset; Z_W is fixed to 128. The problem is ill-posed and affords many solutions; if Z_T is set to be 0D dimension, the texture becomes a "bag of colored pixels" which, when deformed (at will) can reconstruct an image. Increasing the dimension of Z_T (4-32D) lets the network generates aligned texture maps and more exact appearance; further increasing Z_T (128-D) reduces the alignment effect.

B.2 Methods for deformation modeling

In this section, we demonstrate the effect of using different warping modules.

We first show additional comparisons between using our proposed *affine* + *integral* warping and a non-rigid warping field directly output from a convolutional decoder for non-rigid deformation modeling (Figure 16).

We visualize the utility of *affine* and *integral* warping modules in our network with face images (Figure 17). We can see that the affine transformation handles global pose variance (Figure. 17-(b)) but not local non-rigid deformation. Our proposed integral warping module aligns the faces in a non-rigid manner (Figure 17-(c)). Incorporating both deformation modules improves the non-rigid alignment (Figure 17-(d)).

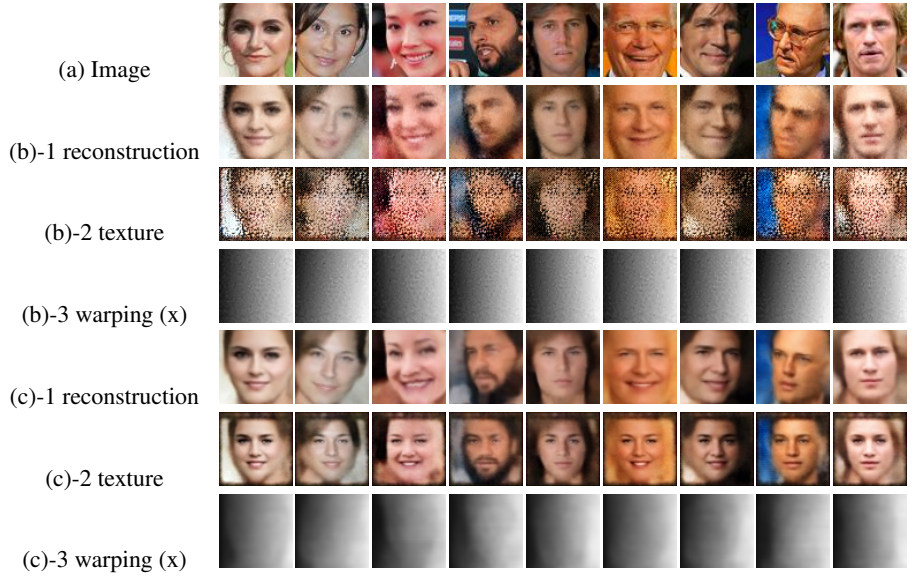

Fig. 16. Comparison between using our proposed *affine* + *integral* warping modules (c) and using a warping field directly predicted from a convolutional decoder (b) for non-rigid deformation modeling. Our non-rigid deformation modeling generates better reconstructions and visually plausible texture maps.

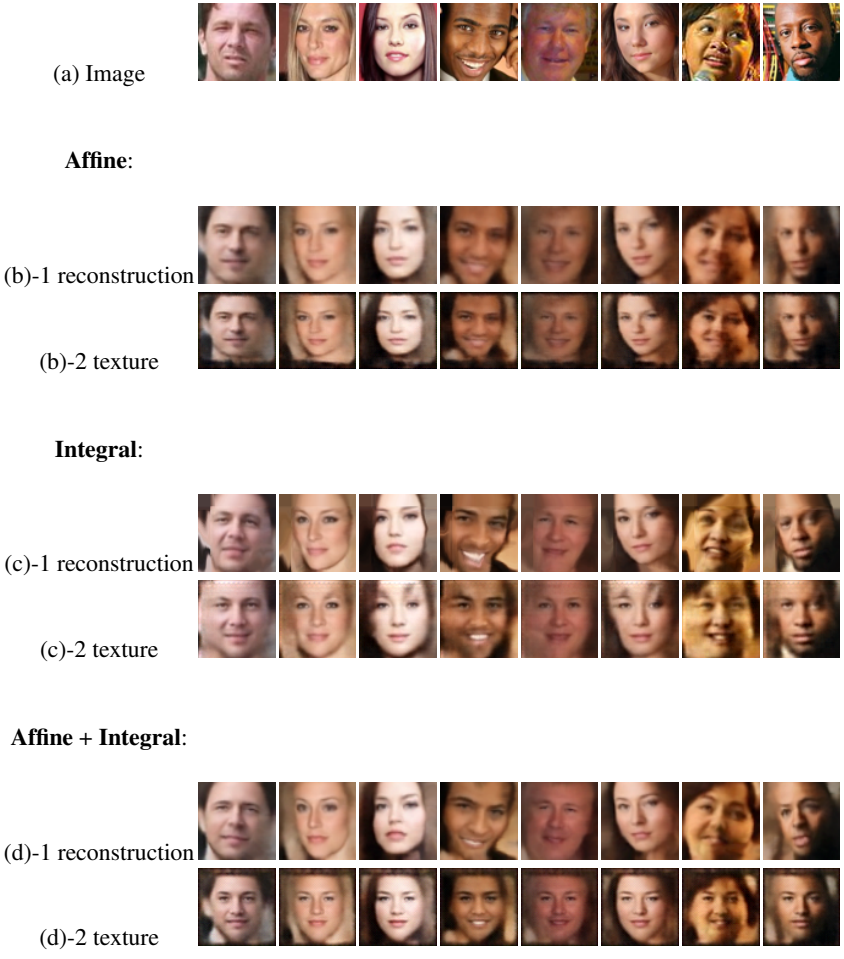

Fig. 17. Effect of affine and integral warping modules using in our network, using faces in-the-wild. The affine transformation can handle global pose variation, as shown in (b) but not local non-rigid deformation- eyes, noses, or other landmarks are not aligned in the decoded texture images. The proposed integral warping module aligns the faces in a non-rigid manner (c), but in an exaggerated manner, causing smears in the texture image, e.g. around eyebrows. Incorporating both deformation modules improves the non-rigid alignment (d). In this experiment, we set $Z_A = 32$, $Z_T = 32$ and $Z_W = 32$.

C Latent Manifold Traversal

We provide additional results and comparisons with a plain autoencoder on traversing the learned manifolds. In addition to Figure 13 in our manuscript, we provide two more sets of results in Figure 18 and Figure 19. Compared to a plain autoencoder, our deforming autoencoder not only generates better reconstructions, but also learns a better face manifold - interpolating between learned latent representations generates sharper and more realistic face images. For this experiment, we use the convolutional encoder and decoder architecture as described in Sec. A.1.

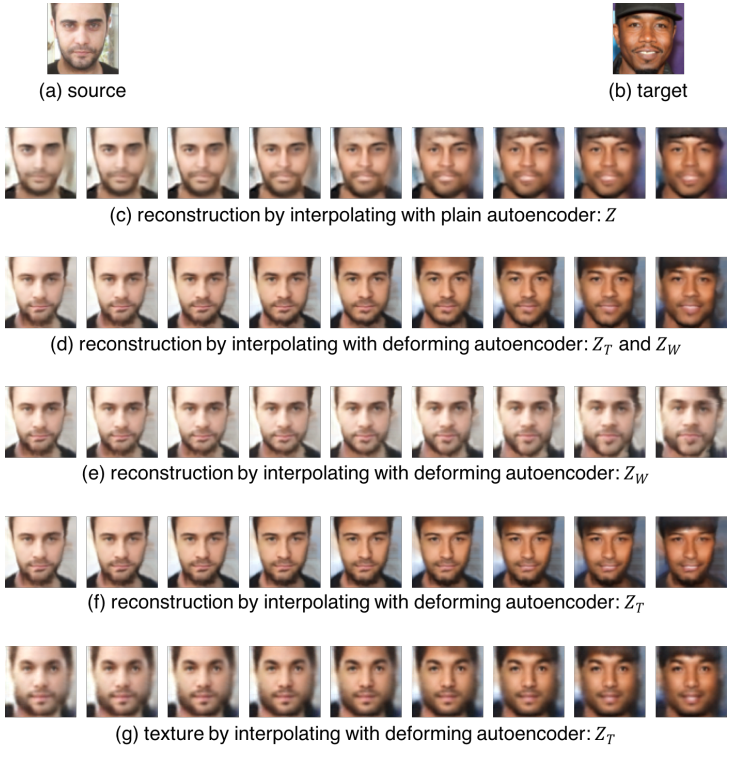

Fig. 18. Interpolating learned representations using networks learned on MAFL dataset. Deforming autoencoder learns better latent representations for face compared to a plain autoencoder. By interpolating the latent representations Z_T and/or Z_W, we observe smooth transition of pose, shape and skin texture. Interpolated results also stays on the face manifold and, generates more realistic image compared to a plain autoencoder.

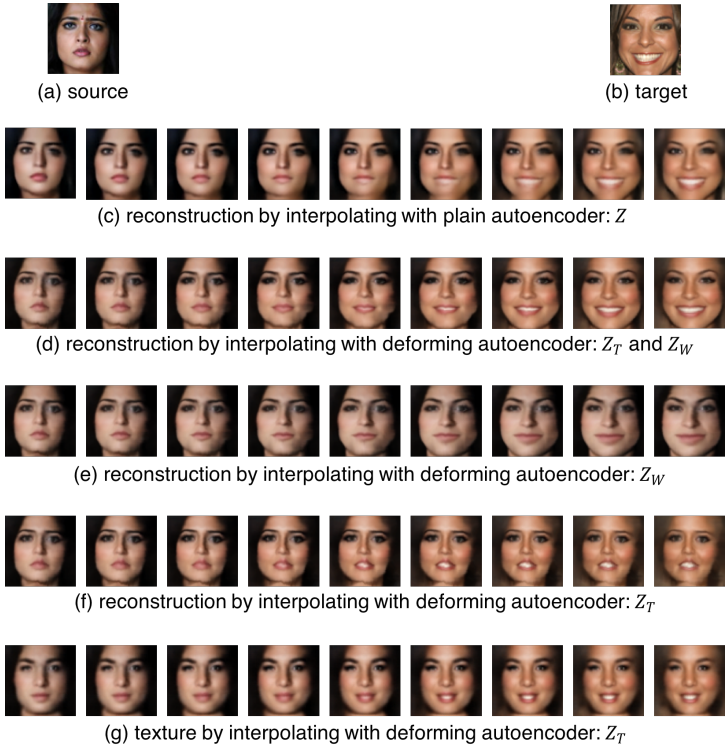

Fig. 19. Interpolating learned representations using networks learned on MAFL dataset. Deforming autoencoder learns better latent representations for face compared to a plain autoencoder. By interpolating the latent representations Z_T and/or Z_W, we observe smooth transition of pose, shape and skin texture. Interpolated results also stays on the face manifold and, generates more realistic image compared to a plain autoencoder.

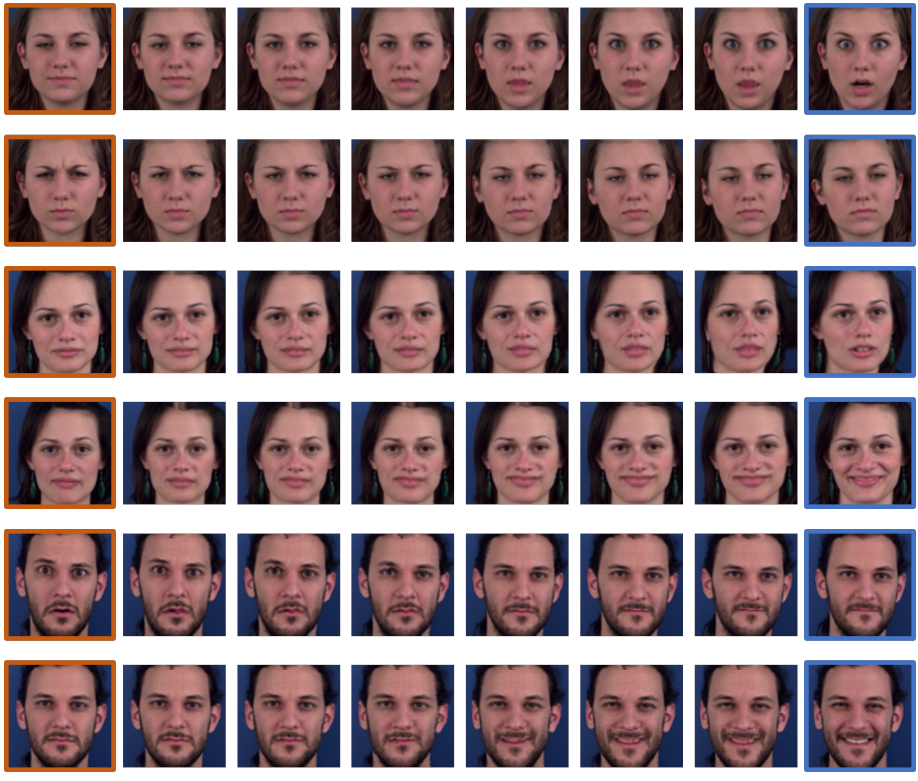

Fig. 20. Expression interpolation: Trained on the MUG facial expression dataset, our network is able to disentangle the facial expression deformation and encode this information in a meaningful latent representation. By interpolating the latent deformation representation from the source (in orange) to the target (in blue), our network generates sharp images and smooth deformation interpolation between expressions as shown in each row. In this experiment, ther model for each subject is independently trained, where we set dimension of Z_T to 0 (assuming single texture for each subject) and dimension of Z_W to 128.

D Intrinsic Decomposition with DAE

In Fig.21 we provide additional results of unsupervised intrinsic disentangling for faces-in-the-wild using Intrinsic-DAE. Using the architecture and objective functions described in Sec. 2.3 of the main paper the network learns to bring faces under different poses and illumination conditions, shown in Fig. 21-(a), to a canonical view, as shown in Fig. 21-(d), while separating the shading, shown in Fig. 21-(b) and albedo, shown in Fig. 21-(c) components in the canonical view using two independent decoders. With the learned deformation from the deformation decoder, we can warp the aligned shading and aligned albedo to its original view as in the input image, as shown in Fig. 21-(e,f).

In Fig. 22, we provide additional results for "changing lighting direction" of a face image using Intinsic-DAE. We show that even without explicitly modeling of geometry, we can simulate smooth and reasonable lighting direction changes in the image by interpolating the learned latent representation for shading, as shown in Fig. 22-a-(4),b-(4).

For Intrinsic-DAE, we use the DenseNet architecture as the encoders and decoders (A.2). The network is trained with a subset of $200,000$ images in the CelebA dataset. The dimensions of latent representations are: 16 for albedo, 16 for shading, and 128 for deformation field.

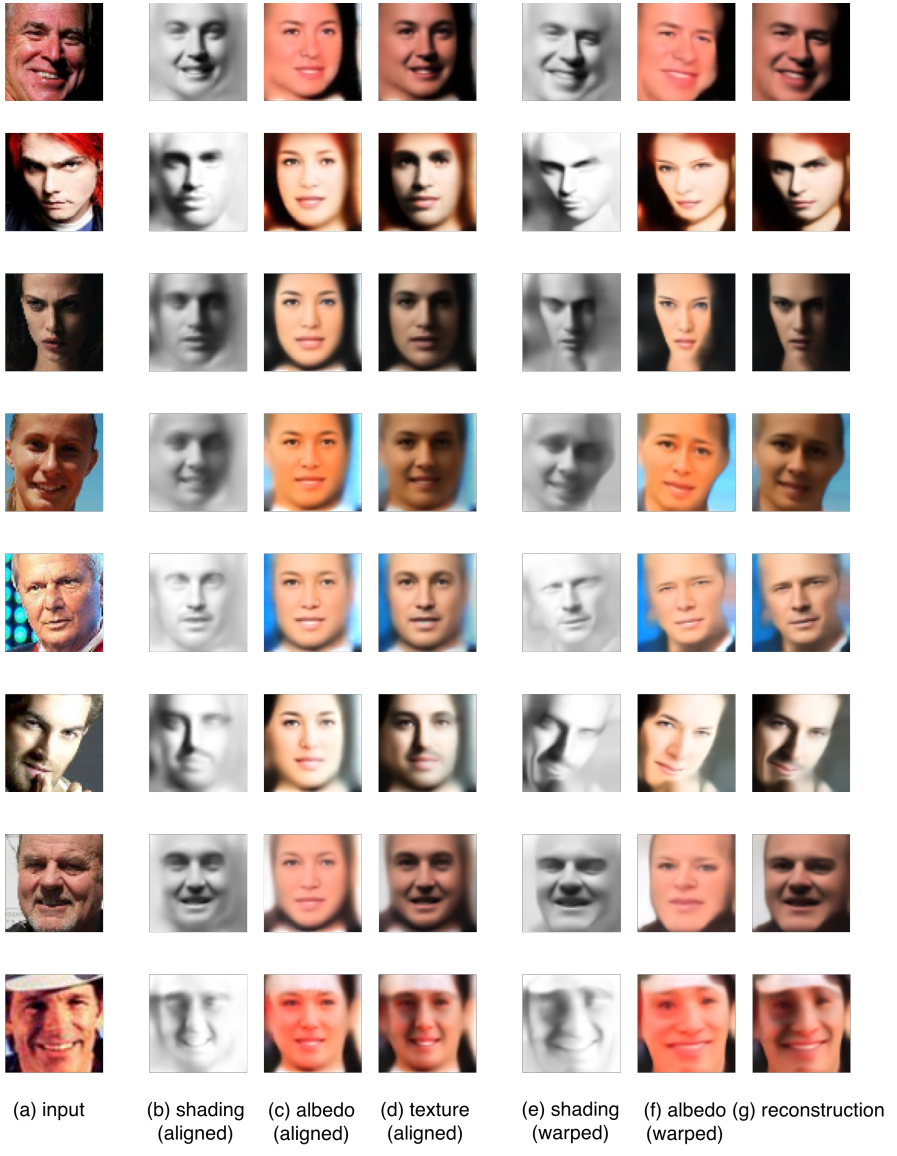

Fig. 21. Unsupervised intrinsic decomposition of faces-in-the-wild using Intrinsic-DAE: The network learns to bring faces under different poses and illumination conditions (a) to a canonical view (d), and further separate the shading (b) and albedo (c) component in the canonical view using two independent decoders. With the learned deformation from the deformation decoder we can warp the aligned shading and aligned albedo to its original view as in the input image (e,f).

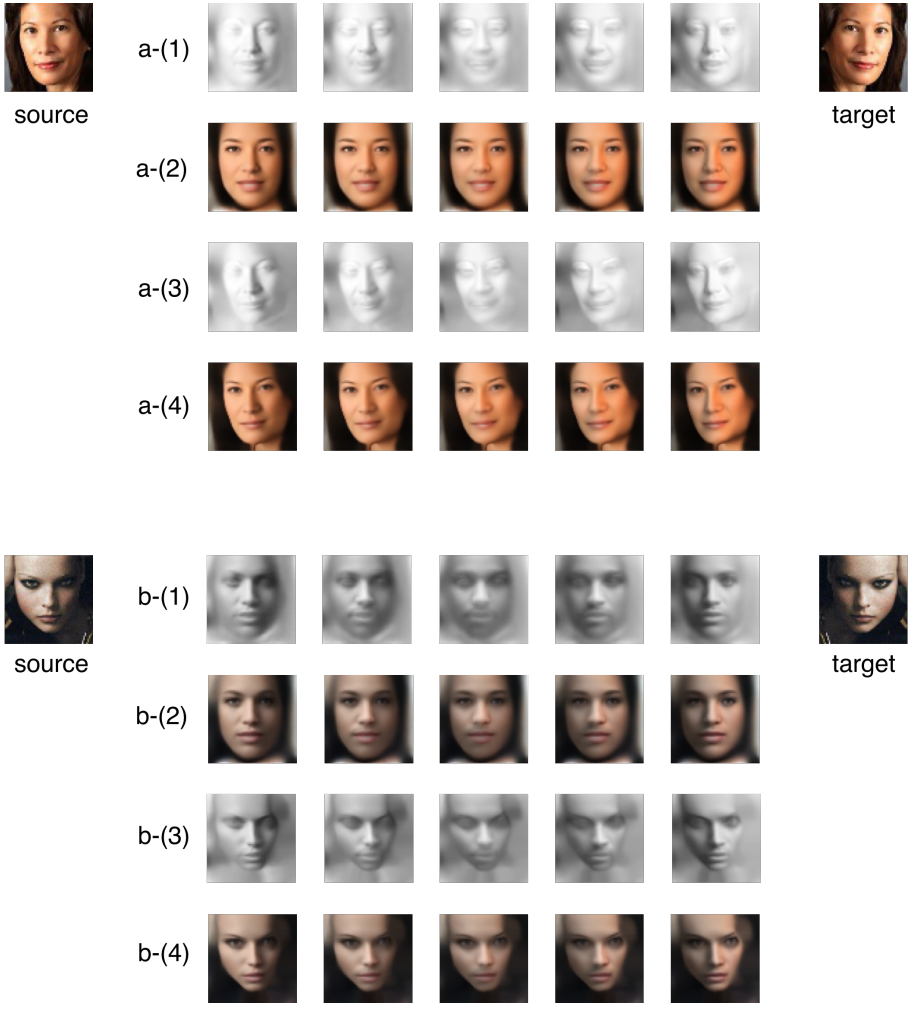

Fig. 22. Lighting manipulation by interpolating latent representation of shading: Intrinsic-DAE allows us to disentangle a latent representation for shading for a given face image in an unsupervised manner. Therefore, manipulating the shading component will result in lighting effects in the output images. In this experiment, we interpolate the latent representation of shading from source to target, which is the mirror of the source with reversed lighting direction. In the result, we can observe that, even without explicitly modeling geometry in our network, we can simulate smooth lighting direction change in both the shading (a-(3), b-(3)) and the final reconstruction (a-(4), b-(4)).